



Temas Selectos de Procesos Químicos-Análisis de Datos con Python

Trimestre 25-O, Clave: 2122075, Grupo: CH01

Profesores: Dr. José de Jesús Álvarez Ramírez (jjar@xanum.uam.mx)
M. en C. Abigail Marin López (amarin@xanum.uam.mx)

Horario: Lunes (AT-219), Miércoles (AT-106) y Viernes (C-105) de 12:00 a 13:30.

Tareas



temas.datosiq@gmail.com





Programación en Python



Octubre 13, 2025

Análisis de Datos con Python



¿Qué es un lenguaje de programación?

Es un conjunto de reglas y símbolos para dar instrucciones a la computadora.

- Funciona como un “idioma” entre humanos y máquinas.

Este lenguaje usa reglas (léxicas, sintácticas y semánticas) que deben cumplirse para que la maquina lo interprete correctamente.

La computadora ejecuta estas instrucciones para realizar tareas específicas, como crear aplicaciones, controlar el funcionamiento de dispositivos o gestionar datos.

¿Qué es Python?

Lenguaje de programación de uso amplio en:

- Aplicaciones web
- Desarrollo de software
- Ciencia de datos
- Machine Learning

Beneficios

1. Sintaxis sencilla y legible, parecida al inglés.
2. Permite programar con **menos líneas de código** que otros lenguajes.
3. Incluye una **gran biblioteca estándar** con código reutilizable.
4. Tiene una **comunidad global activa** que brinda soporte.
5. Es **multiplataforma**: funciona en Windows, macOS, Linux.

Python fue creado a finales de los años ochenta por Guido van Rossum en Stichting Mathematisch Centrum (CWI), en los Países Bajos.



Análisis de Datos

- El análisis de datos es el proceso de inspeccionar, limpiar, transformar y modelar datos con el objetivo de descubrir información útil, llegar a conclusiones y respaldar la toma de decisiones.

Responder
preguntas como:



Generar gráficos, resúmenes estadísticos y patrones descriptivos.

Ciencia de Datos

Responder preguntas como: ¿Qué va a pasar? y ¿Qué debo hacer al Respecto?

Crear modelos predictivos, sistemas de recomendación, algoritmos de clasificación, etc.

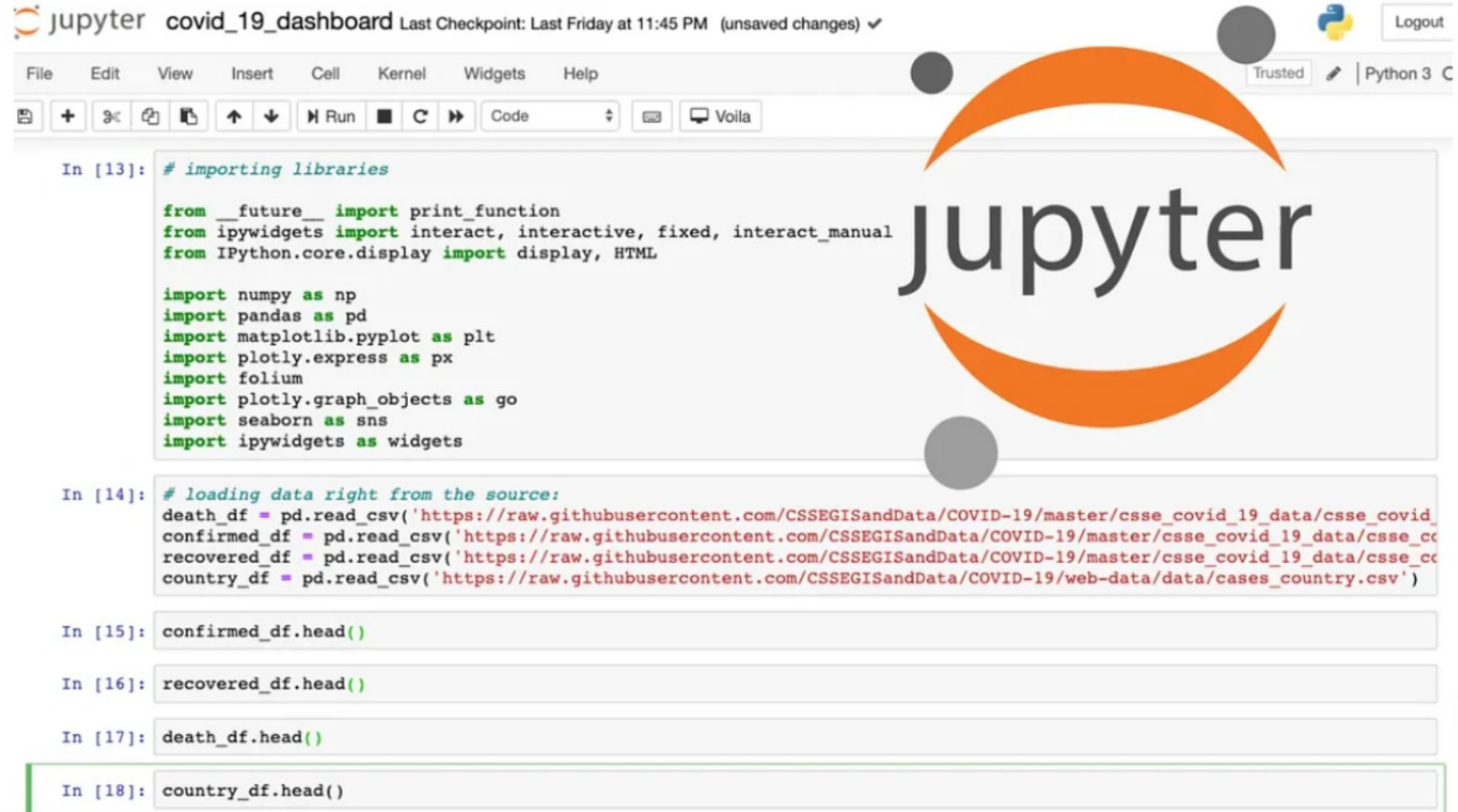


Jupyter Notebook

Aplicación web de código abierto para crear y compartir código y documentos.

Entorno interactivo que permite experimentar y mostrar resultados.

Originalmente pensado para Julia, Python y R



```
In [13]: # importing libraries
from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/death_df.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/confirmed_df.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/recovered_df.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```

Jupyter Notebook



localhost:8888/notebooks/Dropbox/Análisis de Datos en Python/clase 1/Untitled.ipynb? 133% ☆



jupyter **Untitled** (unsaved changes)



Logout

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Trusted

Python 3 (ipykernel) ○



Run



Code



In []:

1



Google Colab

Colab es un servicio de Cuaderno de Jupyter que no requiere configuración para usar y proporciona acceso gratuito a los recursos informáticos.

Adecuado para el aprendizaje automático, la ciencia de datos y la educación.






Funciona en **Mac, Windows o Linux** (solo necesitas un navegador como Chrome o Safari).

No requiere instalación de Python.

Permite **colaborar con otras personas** (como Google Docs).

Viene con muchas librerías científicas preinstaladas (NumPy, pandas, matplotlib, TensorFlow, etc.).

 Carpeta nueva ^C y, luego, F Subir archivo ^C y, luego, U Subir carpeta ^C y, luego, I Documentos de Google ▶ Hojas de cálculo de Google ▶ Presentaciones de Google ▶ Formularios de Google ▶ Más ▶ Papelera Almacenamiento

2.29 GB de 15 GB
utilizado(s)

partidos conmigo

Personas ▼Modificado ▼

Mejores filtros de spam Ahora Drive mueve automáticamente los archivos de spam a la Papelera. [Más información](#)

asado

 Dibujos de Google Google My Maps Sitios de Google Google Apps Script Google Colaboratory



01

ENTEROS

INT

Números enteros
(+/-)

02

FLOTANTE

FLOAT

Números con
decimales

03

CADENA

STRING

Texto

04

BOOLEANO

BOOL

Valores lógicos
(True/False)

Tipos de Datos

¿Por qué los tipos de datos son importantes?

Los tipos de datos no solo **determinan la clase de valor que una variable puede almacenar**, sino que también **dictan cómo se pueden manipular esos valores y qué operaciones se permiten**.

Enteros y flotantes se utilizan para cálculos matemáticos.



Cadenas de texto se manipulan mediante concatenación y búsqueda de subcadenas.

D6		fx =CONCATENAR(A6," ",B6," ",C6)			
	A	B	C	D	E
1	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO		
2	JOSÉ ANTONIO	MENDOZA	BAEZ	JOSÉ ANTONIO MENDOZA BAEZ	
3	PILAR	ROMERO	VILLA	PILAR ROMERO VILLA	
4	ROSARIO	SÁNCHEZ	ALVARADO	ROSARIO SÁNCHEZ ALVARADO	
5	TANIA	DÍAZ	SALGADO	TANIA DÍAZ SALGADO	
6	VICENTE	BOLAÑOS	FÁBREGAS	VICENTE BOLAÑOS FÁBREGAS	
7					




Estructuras de Datos



LISTA LIST

Colecciones ordenadas y modificables de elementos



ARREGLOS ARRAY

Colecciones de elementos del mismo tipo

[1, 2, 3] ó [a, b, c]



OBJETOS OBJECT

Estructuras de datos complejas que pueden contener múltiples valores y tipos de datos.

01

ARITMÉTICOS

Operaciones matemáticas

+, -, *, /, %, **, //

02

RELACIONALES "COMPARACIÓN"

Devuelven True o False al comparar valores

>, <, ==, >=, <=, !=.

True cuando es correcto y False cuando no.

03

LÓGICOS

Se utilizan para combinar expresiones booleanas y tomar decisiones.

& (AND): Devuelve verdadero si ambos operandos son verdaderos.

| (OR): Devuelve verdadero si al menos uno de los operandos es verdadero.

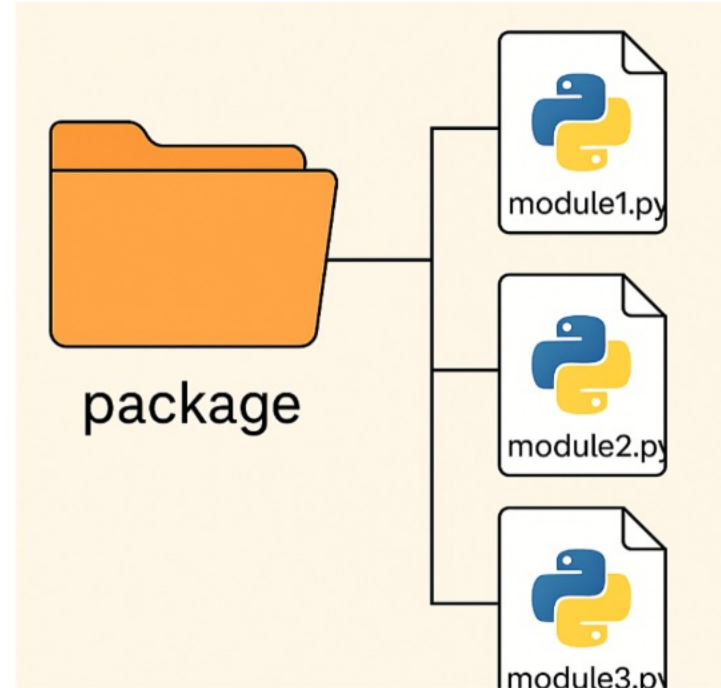
! (NOT): Invierte el valor booleano de un operando.

Tipos de Operadores

Librerías en Python

Una **paquetería (o paquete)** en **Python** es un **conjunto organizado de módulos**, que contienen funciones, clases y variables listas para usar.

Su propósito es **reutilizar código** y evitar que tengas que **programar todo desde cero**.





Tipos de Datos en Python

type():

```
print(type(25))    # <class 'int'>
print(type("Hola")) # <class 'str'>
print(type([1, 2, 3])) # <class 'list'>
```

Listas

```
# Lista vacía  
lista_vacia = []
```

```
# Lista de números  
numeros = [1, 2, 3, 4, 5]
```

```
# Lista de textos  
colores = ["rojo", "azul", "verde"]
```

```
# Lista mixta (diferentes tipos)  
mixta = [10, "hola", True, 3.14]
```

```
# Incluso puedes tener listas dentro de listas  
matriz = [  
    [1, 2, 3],  
    ["a", "b", "c"],  
    [True, False]  
]
```

```
# Lista con diferentes tipos mezclados  
datos = [25, "Juan", True, 3.14, "México"]
```

```
print(datos[0]) # 25 (número entero)  
print(datos[1]) # Juan (texto)  
print(datos[2]) # True (booleano)  
print(datos[3]) # 3.14 (flotante)
```



Arrays

```
import numpy as np

# Arreglo 1D (vector)
vector = np.array([1, 2, 3, 4])
print("Vector:", vector)

# Arreglo 2D (matriz)
matriz = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])
print("Matriz:\n", matriz)
```